221601207
PRACTICALS ON DHTML

Prof. Nasrin Aasofwala
Prof. Neha Samsir

Introduction to JavaScript

Introduction

- JavaScript is a **scripting language** most often used for client-side web development.

JavaScript is an implementation of the **ECMAScript (European Computer Manufacturers Association)** standard.

  - The ECMAScript only defines the syntax/characteristics of the language and a basic set of commonly used objects such as Number, Date, Regular Expression, etc.

- The JavaScript supported in the browsers typically support additional objects.
  - e.g., Window, Frame, Form, DOM object, etc.

JavaScript / JScript

- JavaScript allows for interactivity.
- JavaScript was **developed by Netscape** as a simple programming language (often referred to as a scripting language).
- It is easy to learn and small sections of JavaScript can be added to a web page rather than needing to develop complicated programs.
- It is often used to respond to user actions such as mouse clicks.
- Different brands or/and different versions of browsers may support different implementation of JavaScript.
  - They are not fully compatible
- **JScript** is the Microsoft version of JavaScript.

What can we do with JavaScript?

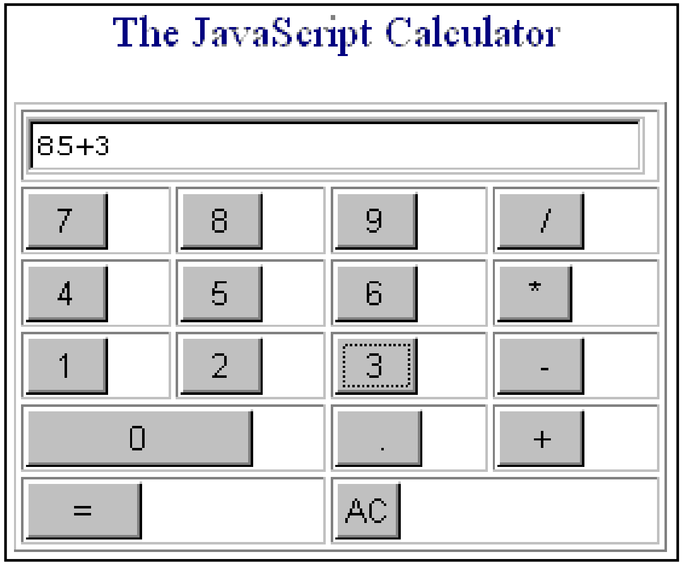
- To create interactive user interface in a web page (e.g., menu, pop-up alert, windows, etc.)

- Manipulating web content dynamically
  - Change the content and style of an element
  - Replace images on a page without page reload
  - Hide/Show contents

- Generate HTML contents on the fly
- Form validation
- AJAX etc…

JavaScript Allows Interactivity

- Improve appearance
  - Especially graphics
  - Visual feedback
- Site navigation
- Perform calculations
- Validation of input
- Other technologies



The JavaScript Calculator

85+3

Features of JavaScript

- **Embedded within HTML page**
  - JavaScript is embedded/included within HTML. You can often see JavaScript in the source of a web page or it is provided for information on the page.
- **Executes on client**
  - JavaScript is mainly used as a client-side language - it downloads with the web page. Once the page has downloaded and is on the users' machine, it is actually the web browser which then interprets the JavaScript instructions. JavaScript pages run quickly, you are not relying on an internet connection to a web server.
- **Simple programming statements combined with HTML tags**
  - Short pieces of JavaScript can be combined with HTML without the need to develop a fully blown program.

Features of JavaScript

- Interpreted (not compiled)
  - There are two types of computer language,
    - **Compiled**
    - **Interpreted**
  - To write or edit a compiled language requires a special piece of software called a compiler.
  - JavaScript belongs to the other category, called interpreted. In the case of JavaScript, this interpretation is done by the browser software at run-time.
  - Because JavaScript is interpreted, this means that no special tools are required to write or edit JavaScript, just a normal text editor. JavaScript web pages can be platform independent i.e. they will run on different browsers and computers (as long as the browser is JavaScript enabled). If you see a JavaScript web page that you like, you may be able to take that JavaScript and use it for your own purposes.

Advantages of JavaScript

- An Interpreted Language
- Embedded Within HTML
- Minimal Syntax – Easy to Learn
- Quick Development
- Designed for Simple, Small Programs
- Performance
- Procedural Capabilities
- Designed for Programming User Events
- Easy Debugging and Testing
- Platform Independence/ Architecture Neutral

Disadvantages of JavaScript

- **Security:** Because the code executes on the users' computer, in some cases it can be exploited for malicious purposes. This is one reason some people choose to disable JavaScript.

- **Reliance on End User:** JavaScript is sometimes interpreted differently by different browsers. Whereas server-side scripts will always produce the same output, client-side scripts can be a little unpredictable. Don't be overly concerned by this though - as long as you test your script in all the major browsers you should be safe

Types of JavaScript

- **External Java Script :** Java Scripts can reside in a separate page.

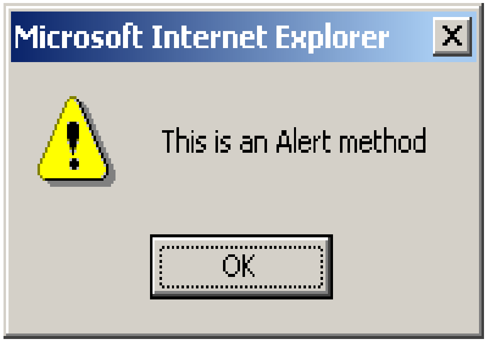
- **Internal JavaScript :** JavaScript can be embedded in HTML documents -- in the <head>, in the <body>, or in both.

- **In line JavaScript :** JavaScript object attributes can be placed in HTML element tags.
  - e.g., <body onLoad="alert('WELCOME')">

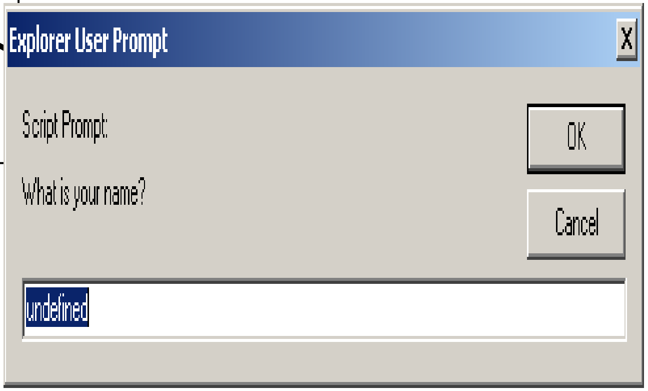
Embedding JavaScript in HTML

- The scripts inside an HTML document is interpreted in the order they appear in the document.
    - Scripts in a function is interpreted when the function is called.
- So where you place the <script> tag matters.

```
<script type="text/javascript">
alert("This is an Alert method");
confirm("Are you OK?");
prompt("What is your name?");
prompt("How old are you?","20");
</script>
```



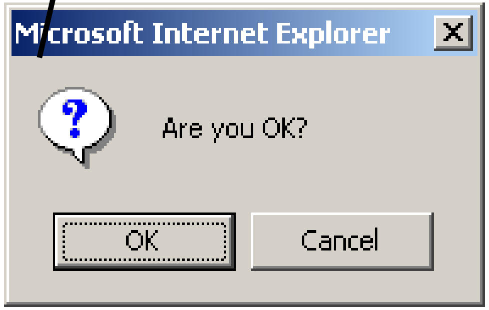
This is an Alert method



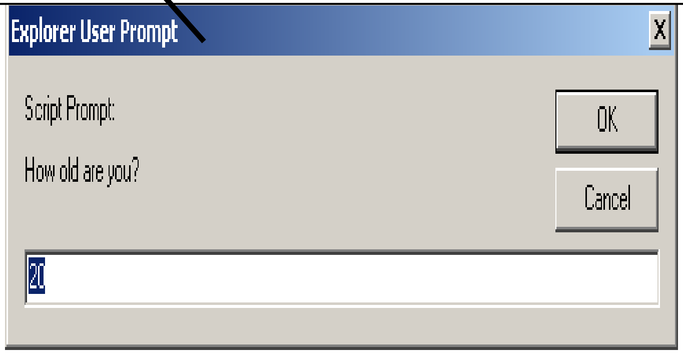
Script Prompt:

What is your name?

undefined



Are you OK?



Script Prompt:

How old are you?

20

DEMO_1.html, DEMO_2.html, DEMO_3.html

Using Separate JavaScript Files (External)

- Linking can be advantageous if many pages use the same script.
- Use the src attribute to include JavaScript codes from an external file.
- The included code is inserted in place.

```
<html>
<head><title>First JavaScript Program</title></head>
<body>
<script type="text/javascript“
        src="your_source_file.js"></script>
</body> </html>
```

Inside your_source_file.js

```
document.write("<hr>");
document.write("Hello World Wide Web");
document.write("<hr>");
```

DEMO_4.html

JavaScript Syntax

- Unlike HTML, JavaScript is case sensitive.
- Dot Syntax is used to combine terms.
  - e.g., document.write("Hello World")
- Certain characters and terms are reserved.
- JavaScript is simple text (ASCII).

Using Comment Tags

- HTML comment tags should bracket any script.
- The <!-- script here --> tags hide scripts in HTML and prevent scripts from displaying in browsers that do not interpret JavaScript.
- Double slashes // are the signal characters for a JavaScript single-line comment.

Java Script Variables

- JavaScript variables are containers for storing data values.

  Example:

  ```
  var x = 5;
  var y = 6;
  var z = x + y;
  ```

DEMO_5.html

Java Script Identifiers

- All JavaScript variables must be identified with unique names.
- These unique names are called identifiers.

Identifiers can be short names (like x and y), or more descriptive names (age, sum, totalVolume).

The general rules for constructing names for variables (unique identifiers) are:

- Names can contain letters, digits, underscores, and dollar signs.
- Names must begin with a letter
- Names can also begin with $ and _ (but we will not use it in this tutorial)
- Names are case sensitive (y and Y are different variables)
- Reserved words (like JavaScript keywords) cannot be used as names

Data Types

Primitive data types

- Number: integer & floating-point numbers
- Boolean: true or false
- String: a sequence of alphanumeric characters

DEMO_6.html

Data Types

Composite data types (or Complex data types)

- Object: a named collection of data
- Array: a sequence of values (an array is actually a predefined object)

Special data types

- Null: the only value is "null" – to represent nothing.
- Undefined: the only value is "undefined" – to represent the value of an uninitialized variable

DEMO_6.html

Strings

- A string variable can store a sequence of alphanumeric characters, spaces and special characters.
- A string can be enclosed by a pair of single quotes (') or double quote ("). You can use escaped character sequence to represent special character (e.g.: \", \n, \t).

typeof operator

- var x = "hello", y;
- alert("Variable x value is " + typeof x);
- alert("Variable y value is " + typeof y);
- alert("Variable x value is " + typeof z);

- This unary operator that tells the type of its operand.
  - Returns a string which can be "number", "string", "boolean", "object", "function", "undefined", and "null”.

DEMO_7.html

Null & Undefined

- An undefined value is represented by the keyword "**undefined**".
  - It represents the value of an uninitialized variable.
- The keyword "null" is used to represent “nothing”
  - Declare and define a variable as “null” if you want the variable to hold nothing.
  - Avoid leaving a variable undefined.

DEMO_8.html

Objects

- Objects refers to windows, documents, images, tables, forms, buttons or links, etc.
- Objects should be named.
- Objects have properties that act as modifiers.

Properties

- Properties are object attributes.
- Object properties are defined by using the object's name, a period, and the property name.
  - e.g., background color is expressed by: document.bgcolor where:
    - document is the object and
    - bgcolor is the property.

Operators & Expression

- Arithmetic Operators
- Logical Operators
- Comparison Operators
- String Operators
- Assignment Operators

Arithmetic Operators

- Arithmetic operators are used to perform arithmetic on numbers (literals or variables).

Operators	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus
++	Return the value then Increment
--	Return the value then Decrement

DEMO_9.html

Comparison Operators

- Comparison operators are used in logical statements to determine equality or difference between variables or values.

Operator	Description
==	Equal
!=	Not Equal
<	Greater Than
>	Less Than
<=	Less than or equal to
>=	Greater than or equal to
===	Strictly equal
!==	Strictly Not equal

DEMO_10.html

Logical Operators

- Logical operators are used to determine the logic between variables or values.

Operator	Description
&&	Logical AND
||	Logical OR
!	Logical NOT

DEMO_11.html

Assignment Operators

Assignment operators assign values to JavaScript variables.

Operator	Description
=	Sets the variable on the left of the = operator to the value of the expression on its right.
+=	Increment the variable on te left of the += operator by the value of expression on its right.
-=	Decrement the variable on te left of the -= operator by the value of expression on its right.
*=	Multiply the variable on te left of the *= operator by the value of expression on its right.
/=	Divides the variable on te left of the /= operator by the value of

String Operators

- The **+** operator can also be used to add (concatenate) strings.
- When used on strings, the + operator is called the concatenation operator.
- The **+=** assignment operator can also be used to add (concatenate) strings.

DEMO_13.html

Conditional Expression : Ternary Operator

- The conditional expression operator is a ternary operator since it takes three operands,

condition ? Value 1 : value 2

- A condition to evaluated and two alternative values to be returned based on the truth or falsity of the condition.

DEMO_14.html

Function in Java Script

- Function are **blocks of Java Script code** that perform a specific task and often return value.
- Function are two types:
  - Built – in Function
  - User Define Function

Built – in Function

- Java script provides several built-in functions that can be used to perform explicit type conversions.
- Eval() - used to convert a string expression to a numeric value.
- parseInt() - used to convert a string value to an integer.
- Parsefloat() - return the first floating point number contained in a string or 0 if the string does not begin with a valid floating point number.

DEMO_15.html DEMO_16.html

User Defined Function in Java Script

- Functions offer the ability to group together Java Script program code that performs a specific task into a single unit that can be used repeatedly whenever required in a Java Script program.
- User defined function
  - Need to be declared
  - Coded
  - Invoked
  - Can be return value

User Define Function in Java Script

- Functions are declared and created using the “***function***” keyword
- A function can comprise of following:
  - A name of the function
  - A list of parameters that will accept values passed to the function when called.
  - A block of Java script code that defines what the function does.
- Syntax:

```
function function_name (parameter1, parameter2, ….)
{
   …....... // block of java script code
}
```

- function_name is case sensitive, can include (_) and has to start with a latter.

DEMO_17.html

Function with parameter passing and returnin value

- Function can be **declared anywhere** within an HTML file.
- Function can be called **by function name**.
- Function can **accept parameter/s** and also **return value**.

DEMO_18.html DEMO_19.html

Dialog Box alert() and confirm()

```
alert("Text to be displayed");
```

- Display a message in a dialog box.
- The dialog box will block the browser.

```
var answer = confirm("Are you sure?");
```

- Display a message in a dialog box with two buttons: "OK" or "Cancel“.
- confirm() returns **true if the user click "OK". Otherwise it returns false.**

DEMO_23.html DEMO_24.html

prompt()

```
prompt("What is your student id number?");
prompt("What is your name?”, "No name");
```

- Display a message and allow the user to enter a value
- The second argument is the "default value" to be displayed in the input textfield.
- Without the default value, "undefined" is shown in the input textfield.
- If the user click the "OK" button, prompt() returns the value in the input textfield as a string.
- If the user click the "Cancel" button, prompt() returns null.

HTML DOM

- An HTML page is redered in a browser.
- The browser assembles all the elements (objects) contained in the HTML page, download from the web server, in its memory.
- Once done the browser then renders these objects in the browser window.
- Once the HTML page is rendered in the browser window, the browser can no longer recognize individual HTML element
- Java Script enabled browsers are capable of recognizing individual objects in an HTML page, after the page has been rendered in the browser
- The Java script enabled browser recognizes and uses the Document Object Model (DOM)

HTML DOM

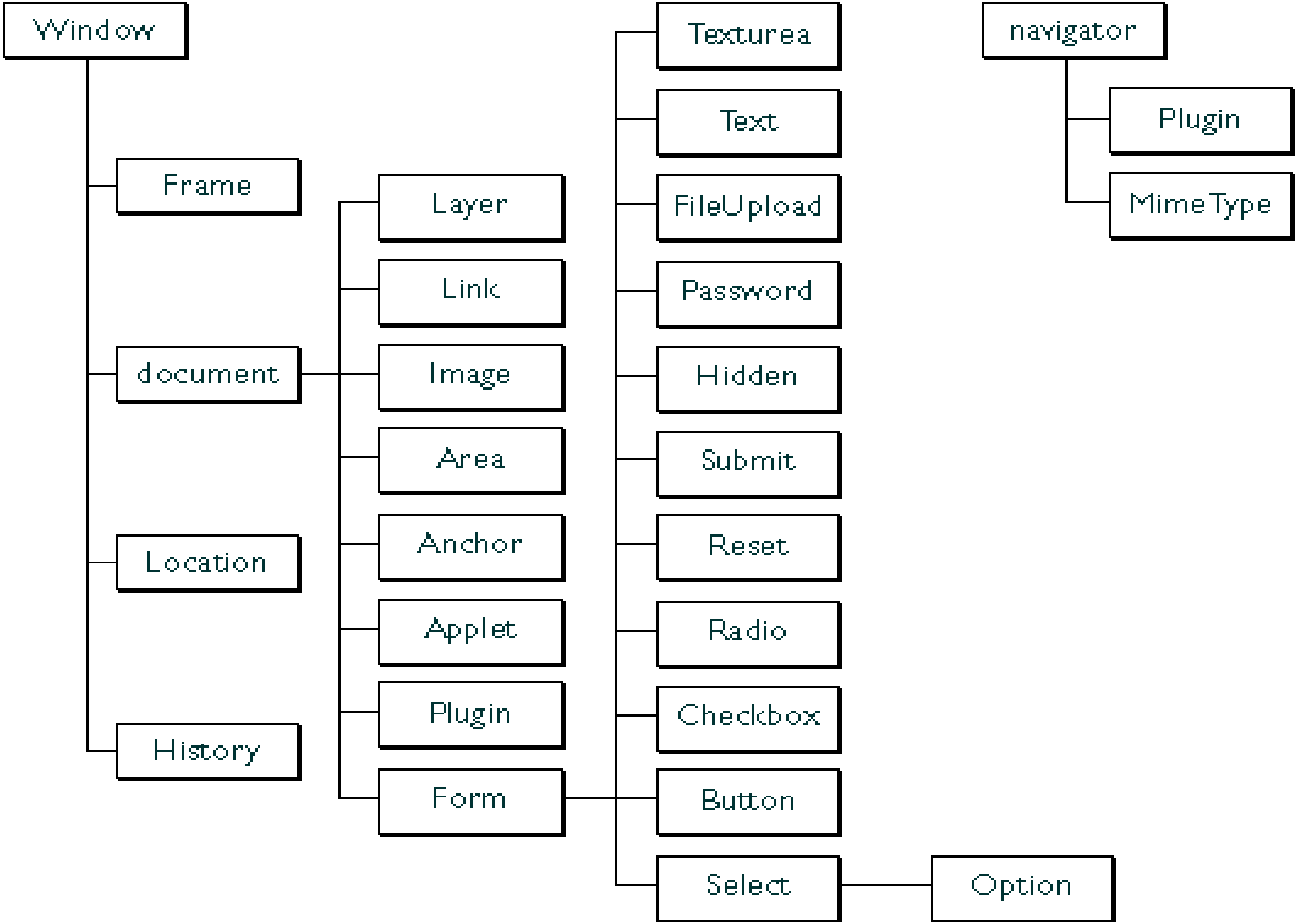
- The HTML objects , which belong to the DOM, have a desending relationship with each other.

- The topmost object in the DOM is **Navigator (the Browser)**
- The next level in the DOM is the browser's **Window**
- The next level in the DOM is the Document
- **Browser**
  - **Window**
    - **Document**
      - **Anchor**
      - **Link**
      - **Form**
        - Textbox
        - Textarea
        - Radiabutton
        - Checkbutton
        - ....

HTML DOM

- DOM stands for the Document Object Model.
- The HTML DOM is the Document Object Model for HTML.
- The HTML DOM defines a standard set of objects for HTML, and a standard way to access and manipulate HTML objects.
- The HTML DOM is a platform and language independent API (application program interface) and can be used by any programming language
- The HTML DOM is used to manipulate HTML documents
- We will use Javascript to interface with the HTML DOM

HTML DOM Hierarchy



- Window
  - Frame
  - document
    - Layer
    - Link
    - Image
    - Area
    - Anchor
    - Applet
    - Plugin
    - Form
      - Texturea
      - Text
      - FileUpload
      - Password
      - Hidden
      - Submit
      - Reset
      - Radio
      - Checkbox
      - Button
      - Select
        - Option
  - Location
  - History
- navigator
  - Plugin
  - MimeType

HTML DOM Example

- The document object is the parent of all the other objects in an HTML document.
- The dot operator is used to represent the parent/child relationship
- The document.body object represents the <body> element of the HTML document.
- The document object is the parent of the body object, and the body object is a child of the document object.
- HTML document objects can have properties (also called attributes). Properties are referenced with the dot operator.
- The document.body.bgColor property defines the background color of the body object.
- The statement :   document.body.bgColor="yellow"
- in the example above, sets the background color of the HTML document to yellow.

HTML DOM Example

- HTML document objects can have methods, subprograms to perform standard tasks.

- Methods are invoked using the dot operator

- write() is a method of the document object. It is invoked by the expression: document.write()

- HTML document objects can respond to events.

- The onclick="ChangeColor()" attribute of the <body> element in the example above, defines an action to take place when the user clicks on the document.
- The ChangeColor() function in the example above, is a JavaScript function.
- The function will be triggered (started) when an event occurs.
- Here it will be triggered, when a user clicks on the HTML document.

String Objects

- Every string in Java Script is an object.
- The string object has a properties, methods to perform a variety of string operation.

Property	Description
length	An integer value indicating the number of characters in the string

String Objects

- The string object has a properties, methods to perform a variety of string operation.

Method	Description
big()	Surrounds the string with the HTML big tag
blink()	Surrounds the string with the HTMl blink tag
bold()	Surrounds the string with the HTML bold tag
charAt()	Given an index as an argument, return the character at the specified index
italics()	Surrounds the string with the HTML <I> tag
toLowercase()	Make the entire string lowercase.
touppercase()	Mkae the entrie stirng uppercase.

DEMO 25,26,27.html

Math Objects

- The string object has a properties, methods to move beyond simple arithmetic manipulations offered by arithmetic operators.

Properties	Description
E	Euler's constant – the base of natural logarithms.
LN10	The natural logarithms of 10(roughly 2.302)
LN2	The natural logarithms of 2 (roughly 0.693)
PI	The ratio of the circumference of a circle to the daimeter of the same circle

Method	Description
abs()	Calculates the absolute value of a number.
ceil()	Returns the next integer greater than or eual to a number
cos()	Calculat the cosine of a number
floor()	Returns the next integer less than or equal to a number
pow()	Calculates the value of onenumber tothe power of a second number
random()	Returns a random number between 0 to1
sin()	Calculates the sine of number

DEMO_28,29.html

Date Objects

- The string object enables Java Script Programers to create an object that can contain information about a particular date.

Method	Description
getDate()	Returns the day of the monthe as an integer from 1 to 31
setDate()	Sets the day of the monthe based on an integer
getHours()	Return the hours an an integer from 0 to23
setHours()	Sets the hours based on an argument from 0 to 23
getTime()	Returns the number of milliseconds

DEMO_30.html

USER DEFINED OBJECTS

- Java Script permits thecreation ofuser definedobjects.

- A user defined object will also be associated with properties and methods which belong to it.

Properties:

- Student.Name
- Student.Age
- Student.Marks

Method:

- Student.Insert();

DEMO_31,32.html